What is claimed is:

1. A computer program product for enhancing performance of a multithreaded application, said computer program product embodied on a computer-readable medium and comprising:

computer-readable program code means for executing a plurality of worker threads;

computer-readable program code means for receiving a plurality of incoming client requests for connections onto an incoming queue;

computer-readable program code means for transferring each of said received client requests for connections from said incoming queue to a wide queue, said wide queue comprising a plurality of queues wherein each of said queues is separately synchronization-protected; and

computer-readable program code means for servicing, by said plurality of worker threads, said client requests by retrieving selected ones of said client requests from said wide queue.

2. The computer program product according to Claim 1, wherein said computer-readable program code means for transferring further comprises:

computer-readable program code means for placing each of said received client requests on a selected one of said plurality of queues using a First-In, First-Out (FIFO) strategy, wherein said selected one of said plurality of queues is selected using a round-robin approach; and further comprising:

computer-readable program code means for returning said retrieved selected ones of said client requests to said wide queue using said FIFO strategy and said round-robin approach upon completion of said computer-readable program code means for servicing.

1 3. A computer program product for enhancing performance of a multithreaded application,

2 said computer program product embodied on a computer-readable medium and comprising:

3 computer-readable program code means for executing a plurality of worker threads;

4 computer-readable program code means for receiving a plurality of incoming client

5 requests onto a queue, wherein each of said client requests is for a connection to a host;

6 computer-readable program code means for retrieving, by individual ones of said worker

7 threads, a selected one of said client requests from said queue;

8 computer-readable program code means for determining a number of connections to said

9 host to which said connection is requested in said selected client request, wherein said number are

10 those which are currently assigned to one or more of said worker threads;

11 computer-readable program code means for processing said selected client request if said

12 number is less than an upper limit, and for not processing said selected client request otherwise;

13 and

14 computer-readable program code means for returning said processed client request or said

15 not processed client request to said queue.

1 4. The computer program product according to Claim 3, wherein said upper limit is a

2 system-wide value.

1 5. The computer program product according to Claim 3, wherein said upper limit is a value

2 specific to said host to which said connection is requested.

1   6.   The computer program product according to Claim 5, wherein said value is dynamically

2   computed, and further comprising:

3   computer-readable program code means for executing a supervisor thread;

4   computer-readable program code means for monitoring, by said supervisor thread,

5   whether connections to each of said hosts succeed or fail; and

6   computer-readable program code means for decrementing said value when said

7   connections to said host fail.

7.   The computer program product according to Claim 6, further comprising:

          computer-readable program code means for incrementing said value when said

connections to said host succeed.

8.   The computer program product according to Claim 6, wherein said computer-readable

program code means for monitoring further comprises:

          computer-readable program code means for setting, by each of said worker threads, a

4   thread time stamp when said worker thread performs active work;

5   computer-readable program code means for comparing, by said supervisor thread, said

6   thread time stamp for each of said worker threads to a system time, thereby computing an elapsed

7   time for said worker thread; and

8   computer-readable program code means for deactivating said worker thread if said elapsed

9   time exceeds a maximum allowable time.

1 9.   The computer program product according to Claim 3, further comprising:

2   computer-readable program code means for providing information for each of said hosts,

3 said information comprising an address of said host and a plurality of in-use flags;

4   computer-readable program code means for setting a selected one of said in-use flags

5 when a particular worker thread is processing work on said connection to a particular host,

6 wherein said selected one of said in-use flags is associated with said particular worker thread; and

7   computer-readable program code means for resetting said selected one of said in-use flags

8 when said particular worker thread stops processing work on said connection to said particular

9 host; and

10   wherein said computer-readable program code means for determining said number of

11 currently-assigned connections further comprises computer-readable program code means for

12 counting how many of said in-use flags are set.

10.   The computer program product according to Claim 3, wherein said queue is a wide queue

comprised of a plurality of First-In, First-Out (FIFO) queues.

1 11.   A system for enhancing performance of a multithreaded application, comprising:

2   means for executing a plurality of worker threads;

3   means for receiving a plurality of incoming client requests for connections onto an

4 incoming queue;

5   means for transferring each of said received client requests for connections from said

6 incoming queue to a wide queue, said wide queue comprising a plurality of queues wherein each

7   of said queues is separately synchronization-protected; and

8   means for servicing, by said plurality of worker threads, said client requests by retrieving

9   selected ones of said client requests from said wide queue.

1   12.   The system according to Claim 11, wherein said means for transferring further comprises:

2   means for placing each of said received client requests on a selected one of said plurality

3   of queues using a First-In, First-Out (FIFO) strategy, wherein said selected one of said plurality of

4   queues is selected using a round-robin approach; and further comprising:

5   means for returning said retrieved selected ones of said client requests to said wide queue

6   using said FIFO strategy and said round-robin approach upon completion of said means for

7   servicing.

13.   A system for enhancing performance of a multithreaded application, comprising:

1   means for executing a plurality of worker threads;

2   means for receiving a plurality of incoming client requests onto a queue, wherein each of

3   means for receiving a plurality of incoming client requests onto a queue, wherein each of

4   said client requests is for a connection to a host;

5   means for retrieving, by individual ones of said worker threads, a selected one of said

6   client requests from said queue;

7   means for determining a number of connections to said host to which said connection is

8   requested in said selected client request, wherein said number are those which are currently

9   assigned to one or more of said worker threads;

10   means for processing said selected client request if said number is less than an upper limit,

11 and for not processing said selected client request otherwise; and

12 means for returning said processed client request or said not processed client request to

13 said queue.

1 14. The system according to Claim 13, wherein said upper limit is a system-wide value.

1 15. The system according to Claim 13, wherein said upper limit is a value specific to said host

2 to which said connection is requested.

16. The system according to Claim 15, wherein said value is dynamically computed, and

further comprising:

means for executing a supervisor thread;

means for monitoring, by said supervisor thread, whether connections to each of said hosts

succeed or fail; and

means for decrementing said value when said connections to said host fail.

1 17. The system according to Claim 16, further comprising:

2 means for incrementing said value when said connections to said host succeed.

1 18. The system according to Claim 16, wherein said means for monitoring further comprises:

2 means for setting, by each of said worker threads, a thread time stamp when said worker

3 thread performs active work;

4   means for comparing, by said supervisor thread, said thread time stamp for each of said

5   worker threads to a system time, thereby computing an elapsed time for said worker thread; and

6   means for deactivating said worker thread if said elapsed time exceeds a maximum

7   allowable time.

1   19.   The system according to Claim 13, further comprising:

2   means for providing information for each of said hosts, said information comprising an

3   address of said host and a plurality of in-use flags;

4   means for setting a selected one of said in-use flags when a particular worker thread is

5   processing work on said connection to a particular host, wherein said selected one of said in-use

6   flags is associated with said particular worker thread; and

7   means for resetting said selected one of said in-use flags when said particular worker

8   thread stops processing work on said connection to said particular host; and

9   wherein said means for determining said number of currently-assigned connections further

10   comprises means for counting how many of said in-use flags are set.

1   20.   The system according to Claim 13, wherein said queue is a wide queue comprised of a

2   plurality of First-In, First-Out (FIFO) queues.

1   21.   A method for enhancing performance of a multithreaded application, comprising the steps

2   of:

3   executing a plurality of worker threads;

4   receiving a plurality of incoming client requests for connections onto an incoming queue;

5   transferring each of said received client requests for connections from said incoming queue

6   to a wide queue, said wide queue comprising a plurality of queues wherein each of said queues is

7   separately synchronization-protected; and

8   servicing, by said plurality of worker threads, said client requests by retrieving selected

9   ones of said client requests from said wide queue.

1   22.   The method according to Claim 21, wherein said transferring step further comprises the

2   steps of:

3   placing each of said received client requests on a selected one of said plurality of queues

4   using a First-In, First-Out (FIFO) strategy, wherein said selected one of said plurality of queues is

5   selected using a round-robin approach; and further comprising the step of:

6   returning said retrieved selected ones of said client requests to said wide queue using said

7   FIFO strategy and said round-robin approach upon completion of said servicing step.

1   23.   A method for enhancing performance of a multithreaded application, comprising the steps

2   of:

3   executing a plurality of worker threads;

4   receiving a plurality of incoming client requests onto a queue, wherein each of said client

5   requests is for a connection to a host;

6   retrieving, by individual ones of said worker threads, a selected one of said client requests

7   from said queue;

8   determining a number of connections to said host to which said connection is requested in

9   said selected client request, wherein said number are those which are currently assigned to one or

10   more of said worker threads;

11   processing said selected client request if said number is less than an upper limit, and not

12   processing said selected client request otherwise; and

13   returning said processed client request or said not processed client request to said queue.

1   24.   The method according to Claim 23, wherein said upper limit is a system-wide value.

25.   The method according to Claim 23, wherein said upper limit is a value specific to said host
to which said connection is requested.

26.   The method according to Claim 25, wherein said value is dynamically computed, and
further comprising the steps of:

      executing a supervisor thread;

4   monitoring, by said supervisor thread, whether connections to each of said hosts succeed

5   or fail; and

6   decrementing said value when said connections to said host fail.

1   27.   The method according to Claim 26, further comprising the step of incrementing said value

2   when said connections to said host succeed.

1 28. The method according to Claim 26, wherein said monitoring step further comprises the

2 steps of:

3 setting, by each of said worker threads, a thread time stamp when said worker thread

4 performs active work;

5 comparing, by said supervisor thread, said thread time stamp for each of said worker

6 threads to a system time, thereby computing an elapsed time for said worker thread; and

7 deactivating said worker thread if said elapsed time exceeds a maximum allowable time.

29. The method according to Claim 23, further comprising the steps of:

providing information for each of said hosts, said information comprising an address of

said host and a plurality of in-use flags;

setting a selected one of said in-use flags when a particular worker thread is processing

work on said connection to a particular host, wherein said selected one of said in-use flags is

associated with said particular worker thread; and

resetting said selected one of said in-use flags when said particular worker thread stops

8 processing work on said connection to said particular host; and

9 wherein said step of determining said number of currently-assigned connections further

10 comprises counting how many of said in-use flags are set.

1 30. The method according to Claim 23, wherein said queue is a wide queue comprised of a

2 plurality of First-In, First-Out (FIFO) queues.